ABC201题解

A

直接判断就行了。

B

结构体排序之后输出第二个。

C

最简单粗暴的方法是直接枚举0000~9999然后逐一判断是否满足字符串条件即可。

D

一个DP。

状态：dp[i][j]为走到(i,j)时两人的分差（先手-后手）。

先手方要最大化这个值，后手方要最小化这个值。注意到先手方走到的坐标和一定是奇数，后手方走到的坐标和一定是偶数，那么方程可以写成：

当i+j为奇数时，dp[i][j]=max(dp[i-1][j]+a[i][j],dp[i][j-1]+a[i][j])

当i+j为偶数时，dp[i][j]=min(dp[i-1][j]-a[i][j],dp[i][j-1]-a[i][j])

但是正序遍历有后效性，所以我们要倒序遍历：

当i+j为奇数时，dp[i][j]=min(dp[i+1][j]-a[i+1][j],dp[i][j+1]-a[i][j+1])

当i+j为偶数时，dp[i][j]=max(dp[i+1][j]+a[i+1][j],dp[i][j+1]+a[i][j+1])

E

首先，容易证明dis(i,j)=dis(i,x) xor dis(x,j)

这个式子告诉我们，无论以哪个点作为树的根，树上两点之间的最短路径的边权的异或和等于两点到根的最短路径的边权的异或和的异或。（感性理解就是公共部分被异或了两次消掉了）

那么不妨设 1 为根，先 dfs 一遍求出根到每个点的异或和，再考虑统计答案。

由于异或的每一位是独立的，因此不妨对每一位考虑对答案的贡献。

注意到，只有 0⊕1=1 会对答案产生贡献，因此可以得出答案的计算式：

ans=∑2ipi(n−pi)

其中，pi表示在所有的异或和中，第 i 位是 1 的数量，n−pi就是第 i 位是 0 的数量，根据乘法原理，总贡献就是 2ipi(n−pi)。

时间复杂度为 O(nlogV)，其中 V 为异或和的值域。

F

首先一个人最多移动一次，反复横跳是没有意义的。

我们考虑那些没有移动的人，显然至少存在一个。证明只需考虑把编号最大的一个向左移动的固定住，显然至少可以减少这一个人的代价。

那我们不难发现，不移动的人相当于把原来的人按照编号划分成了若干个区间。具体地说，在固定的人确定为 S1<S2<...<Sk后，你的策略应该是：

编号小于 S1的从大到小放到左边，代价为 min(Ai,Bi)；

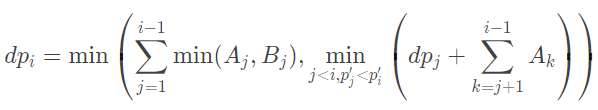
编号大于 Sk的从小到大放到右边，代价为 min(Ai,Ci)；

其余的放到对应的位置上，代价为 Ai。

问题变成随意安排一个人是否不动，最小化代价和。因为 S 中的数在原序列中也要升序，所i有从小到大考虑的话，前面人的安排不影响后面的人，即问题没有后效性。

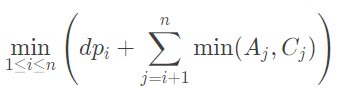
因而我们可以进行dp。设 dp[i]表示考虑到编号不超过 i 的人，且满足 i∈S 的最小代价和。

不难得到转移方程：



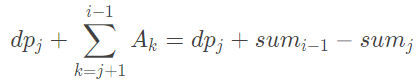
这里p’是原排列的逆，前半部分表示令 i 为第一个固定住的，后半部分表示枚举前一个固定的人 j。

最后答案是：



两个求和都可以用前缀和优化，但后半部分要枚举j，复杂度还是O(n2)过不了。

考虑先把常数分离出来，设sum[i]为Ai的前缀和，则：



将dp[j]-sum[j]作为一个整体，就变成找sum[i-1]的最小值。找前缀min和单点修改的话，树状数组或者线段树都行，复杂度O(n logn)。